UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte Scott N. Gerard

Appeal No. _____
Application No. 10/760,592

APPEAL BRIEF

Attorney Docket No. ROC920030316US1 PATENT
Confirmation No. 1109

CERTIFICATE OF ELECTRONIC TRANSMISSION
I hereby certify that this correspondence for Application No. 10/760,592 is being electronically transmitted to Technology Center 2134 via EFS-WEB, on March 28, 2008.

/Scott A. Stinebruner/ March 28, 2008
Scott A. Stinebruner, Reg. No. 38,323 Date

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Scott N. Gerard	Art Unit: 2134
Application No.:	10/760,592	Examiner: Yonas A. Bayau
Filed:	January 20, 2004	
For:	DISTRIBUTED COMPUTATION IN UNTRUSTED COMPUTING ENVIRONMENTS USING DISTRACTIVE COMPUTATIONAL UNITS	

Mail Stop Appeal Brief - Patents
Commissioner for Patent
P.O. Box 1450
Alexandria, VA 22213-1450

APPEAL BRIEF

I. REAL PARTY IN INTEREST

This application is assigned to International Business Machines Corporation, of Armonk, New York.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences.

III. STATUS OF CLAIMS

Claims 1-4, 6-8, 13-20, 22-24 and 29-33 are pending in the Application, stand rejected, and are now on appeal. Claims 5, 9-12, 21, 25-28 and 34 have been canceled.

IV. STATUS OF AMENDMENTS

There have been no amendments filed subsequent to the final rejection mailed September 26, 2007.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Applicant's invention is generally directed to the use of distractive computational units to inhibit the reconstitution of computations by untrusted party on untrusted computer. The invention may be used, for example, in a grid computing environment, where a collection of potentially heterogeneous and geographically remote computer systems are connected together in a distributed fashion to share their respective resources and present the appearance of a single large and powerful virtual computer system (page 1, lines 13-19).

While some grid computing environments may be maintained within a single organization, in many instances environments of this sort are used by multiple organizations, or may even be publicly maintained and/or supported. Such latter environments, however, raise some significant security issues by virtue of the multi-organizational structure of a grid. (page 2, line 20 to page 3, line 12). In many distributed and computational utility environments, for example, it is anticipated that numerous organizations will provide computing resources to the infrastructure for the environment. In some instances, these organizations may even be competitors of one another. A client wishing to have work performed in such an environment may therefore not have complete control over the confidentiality of its data (page 3, lines 13-17).

While encrypted communications may be used to protect communications between computers, when the underlying processing of data is performed on a particular computer in a computational utility environment, that computer typically must be capable of decrypting the data prior to performing the computation. As a result, it may be possible that computations performed on behalf of a client in a computational utility environment may be performed, at least in part, by computing resources hosted by competitors or other untrusted parties. A substantial risk therefore exists that a curious or malicious untrusted party will eavesdrop on computations and/or communications associated with computations while hosting computing resources,

thereby creating a risk that a client's confidential data may be compromised. (page 3, lines 18-29).

To address this concern, Applicant's invention initiates the execution of distractive computational units along with the execution of other computational units on an untrusted computer to inhibit the reconstitution of a computation by an untrusted party. In addition, computations are partitioned into a plurality of computational units, such that one or more distractive computational units may be generated and supplied to one or more resource providers for execution along with those of the partitioned computation. It is believed that the presence of a distractive computational unit, along with one or more computational units that are generated as a result of partitioning a computation of interest, can significantly complicate detection and reconstitution of the computation or its overall result from the computational units supplied to an untrusted computer (page 5, lines 2-14).

For the convenience of the Board, each of the independent claims has been reproduced below and annotated with references to the specification and drawings to satisfy the requirement to concisely explain the specifically claimed subject matter:

Claim 1

A method of initiating performance of a computation on at least one untrusted computer (page 5, lines 3-5, Fig. 3, block 56), the method comprising:

partitioning the computation into a plurality of computational units that are combinable to generate a result for the computation (page 16, lines 20-23, Fig. 5, block 122), wherein the computation includes a plurality of arguments, wherein partitioning the computation into the plurality of computational units comprises partitioning using the Chinese Remainder Theorem (CRT) (page 16, lines 20-23, Fig. 5, block 122), wherein partitioning the computation into the plurality of computational units comprises selecting a plurality of relatively prime moduli (page 15, line 28 to page 16, line 12, Fig. 5, block 122) and generating each computational unit by performing a modulo operation on each of the plurality of arguments using one of the plurality of relatively prime moduli (page 16, lines 20-23, Fig. 5, block 122), and wherein selecting the plurality of relatively prime

moduli includes selecting each modulus from a superset of relatively prime moduli (page 16, lines 20-23, Fig. 5, block 122);

partitioning a plurality of computations into multiple computational units using different sets of moduli selected from the superset of relatively prime moduli (page 16, lines 20-23, Fig. 5, block 122);

generating at least one distractive computational unit, wherein the distractive computational unit comprises a dummy computational unit (page 14, lines 2-29, Fig. 4, block 104);

initiating execution of both the at least one distractive computational unit and computational units from multiple computations on the untrusted computer to inhibit reconstitution of the computations by an untrusted party (page 14, lines 2-10, Fig. 4, block 106);

receiving result data generated during execution of the computational units from the multiple computations (page 14, lines 2-10, Fig. 4, block 108); and

generating results for the multiple computations from the result data (page 14, lines 2-10, Fig. 4, block 112).

Claim 17

An apparatus (Fig. 1, block 10), comprising:

at least one processor (Fig. 3, block 42); and

program code (page 11, lines 11-29) configured to be executed by the at least one processor to initiate performance of a computation on at least one untrusted computer (page 5, lines 3-5, Fig. 3, block 56) by partitioning the computation into a plurality of computational units that are combinable to generate a result for the computation (page 16, lines 20-23, Fig. 5, block 122), generating at least one distractive computational unit (page 14, lines 2-29, Fig. 4, block 104), and initiating execution of both the at least one distractive computational unit and at least one of the plurality of computational units on the untrusted computer to inhibit reconstitution of the computation by an untrusted party (page 14, lines 2-10, Fig. 4, block 106);

wherein the program code is configured to partition the computation into the plurality of computational units using the Chinese Remainder Theorem (CRT) (page 16, lines 20-23, Fig. 5, block 122), wherein the computation includes a plurality of arguments, wherein the program code is configured to partition the computation into the plurality of computational units by selecting a plurality of relatively prime moduli (page 15, line 28 to page 16, line 12, Fig. 5, block 122), and generating each computational unit by performing a modulo operation on each of the plurality of arguments using one of the plurality of relatively prime moduli (page 16, lines 20-23, Fig. 5, block 122), wherein the program code is configured to select the plurality of relatively prime moduli from a superset of relatively prime moduli (page 16, lines 20-23, Fig. 5, block 122), wherein the program code is further configured to partition a plurality of computations into multiple computational units using different sets of moduli selected from the superset of relatively prime moduli (page 16, lines 20-23, Fig. 5, block 122), and initiate execution of computational units from multiple computations on the untrusted computer (page 14, lines 2-10, Fig. 4, block 106), wherein the distractive computational unit comprises a dummy computational unit (page 14, lines 2-29, Fig. 4, block 104), and wherein the program code is further configured to receive result data generated during execution of the computational units from the multiple computations (page 14, lines 2-10, Fig. 4, block 108) and generate results for the multiple computations from the result data (page 14, lines 2-10, Fig. 4, block 112).

Claim 33

A program product (page 11, lines 11-29), comprising:

program code (page 11, lines 11-29) configured to initiate performance of a computation on at least one untrusted computer (page 5, lines 3-5, Fig. 3, block 56) by partitioning the computation into a plurality of computational units that are combinable to generate a result for the computation (page 16, lines 20-23, Fig. 5, block 122), generating at least one distractive computational unit (page 14, lines 2-29, Fig. 4, block 104), and initiating execution of both the at least one distractive computational unit and at least one of the plurality of computational units on the untrusted computer to inhibit reconstitution of the computation by an untrusted party (page 14, lines 2-10, Fig. 4, block 106); and

a computer readable recordable storage medium (page 11, lines 11-29) bearing the program code;

wherein the program code is configured to partition the computation into the plurality of computational units using the Chinese Remainder Theorem (CRT) (page 16, lines 20-23, Fig. 5, block 122), wherein the computation includes a plurality of arguments, wherein the program code is configured to partition the computation into the plurality of computational units by selecting a plurality of relatively prime moduli (page 15, line 28 to page 16, line 12, Fig. 5, block 122), and generating each computational unit by performing a modulo operation on each of the plurality of arguments using one of the plurality of relatively prime moduli (page 16, lines 20-23, Fig. 5, block 122), wherein the program code is configured to select the plurality of relatively prime moduli from a superset of relatively prime moduli (page 16, lines 20-23, Fig. 5, block 122), wherein the program code is further configured to partition a plurality of computations into multiple computational units using different sets of moduli selected from the superset of relatively prime moduli (page 16, lines 20-23, Fig. 5, block 122), and initiate execution of computational units from multiple computations on the untrusted computer (page 14, lines 2-10, Fig. 4, block 106), and wherein the distractive computational unit comprises a dummy computational unit (page 14, lines 2-29, Fig. 4, block 104), and wherein the program code is further configured to receive result data generated during execution of the computational units from the multiple computations (page 14, lines 2-10, Fig. 4, block 108) and generate results for the multiple computations from the result data (page 14, lines 2-10, Fig. 4, block 112).

It should be noted that, as none of the claims recite any means plus function or step plus function elements, no identification of such elements is required pursuant to 37 C.F.R §41.37(c)(1)(v). Furthermore, there is no requirement in 37 C.F.R §41.37(c)(1)(v) to provide support for the subject matter in the separately argued dependent claims, as none of these claims recite means plus function or step plus function elements, and so no discussion of any of these claims is provided.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1-4, 6-8, 13-20, 22-24 and 29-33 are rejected under 35 U.S.C. § 103 (a) as being obvious over Jakobsson et al. (U.S. Patent No. 6,950,937) (hereinafter "Jakobsson") in view of Elbe et al. (WO 02/48857 A2) (hereinafter "Elbe").

VII. ARGUMENT

Applicant respectfully submits that the Examiner's rejections of claims 1-4, 6-8, 13-20, 22-24 and 29-33 are not supported on the record, and should be reversed. All such claims have been rejected as being obvious by the Examiner in view of Jakobsson and Elbe. Based upon the Supreme Court's decision in *KSR International Co. v. Teleflex Inc.*, 127 S. Ct. 1727, 1734, 82 USPQ2d 1385, 1391 (2007), a *prima facie* showing of obviousness requires that the Examiner establish that the differences between a claimed invention and the prior art "are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art." 35 U.S.C. §103(a). Such a showing requires that all claimed features be disclosed or suggested by the prior art.

Four factors generally control an obviousness inquiry: 1) the scope and content of the prior art; 2) the differences between the prior art and the claims; 3) the level of ordinary skill in the pertinent art; and 4) secondary considerations of non-obviousness, such as commercial success of products covered by the patent claims, a long felt but unresolved need for the invention, and failed attempts by others to make the invention. *KSR*, 127 S. Ct. at 1734 (quoting *Graham v. John Deere Company*, 383 U.S. 1, 17-18 (1966)) ("While the sequence of these questions might be reordered in any particular case, the [Graham] factors continue to define the inquiry that controls.").

Moreover, in *KSR*, the Court explained that "[o]ften, it will be necessary for a court to look to interrelated teachings of multiple patents; the effects of demands known to the design community or present in the marketplace; and the background knowledge possessed by a person having ordinary skill in the art, all in order to determine whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue" and "[t]o facilitate

review, this analysis should be made explicit." *KSR*, 127 S. Ct. at 1740-41 citing *In re Kahn*, 441 F.3d 977, 988, 78 USPQ2d 1329, 1336 (Fed. Cir. 2006) ("[R]ejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness"). Not every combination is obvious "because inventions in most, if not all, instances rely upon building blocks long since uncovered, and claimed discoveries almost of necessity will be combinations of what, in some sense, is already known." *KSR*, 127 S. Ct. at 1741.

As a result, after *KSR*, while there is no rigid requirement for an explicit "teaching, suggestion or motivation" to combine references, there still must be some evidence of "a reason that would have prompted a person of ordinary skill in the relevant field to combine the elements in the way the claimed new invention does" in an obviousness determination. *KSR*, 127 S. Ct. at 1731.

Applicant respectfully submits that, in the instant case, the Examiner has failed to establish obviousness as to claims 1-4, 6-8, 13-20, 22-24 and 29-33, and as such, the rejections thereof should be reversed. Indeed, the combination of Jakobsson and Elbe still does not teach or suggest each and every element of the pending claims, which is a requirement that has not changed after *KSR*.

Applicant's remarks in rebuttal to the Examiner's rejections are presented below, starting with the relevant independent claims. In some cases, specific discussions of particular claims are not made in the interests of streamlining the appeal. The omission of a discussion with respect to any particular claim, however, should not be interpreted as an acquiescence as to the merits of the Examiner's rejection of the claim, particularly with respect to claims reciting features that are addressed in connection with the rejections applied to other claims pending in the appeal.

In addition, Applicant notes at the outset that the arguments presented below are highly similar to those presented in Applicant's last response. In the Final Office Action dated September 26, 2007, the Examiner merely inserted into the rejection of the independent claims, the citations to passages in Jakobsson and Elbe that were previously used in the rejections of the various dependent claims that Applicant had incorporated into the independent claim. The

Examiner then failed to address Applicant's arguments in support of patentability by asserting that the rejections were new, and thus rendered Applicant's arguments moot. Given that the "new" rejections consisted of nothing more than cutting and pasting citations used in the prior rejections of the now-incorporated dependent claims, Applicant submits that these rejections did not in fact render Applicant's arguments moot.

Furthermore, in neither Office Action has the Examiner attempted to address by way of written remarks the teachings of the prior art, the claims, or Applicant's arguments in support of patentability. Applicant submits that the arguments presented in the last response are entitled to a full and fair hearing by the Office, and as such, those arguments have been reiterated below. Should the Examiner choose to introduce arguments rebutting Applicant's arguments in support of patentability, Applicant will address those arguments in a Reply Brief as necessary.

Independent Claim 1

Claim 1 generally recites a method of initiating performance of a computation on at least one untrusted computer. The method comprises partitioning the computation into a plurality of computational units that are combinable to generate a result for the computation, generating at least one distractive computational unit, initiating execution of both the at least one distractive computational unit and at least one of the plurality of computational units on the untrusted computer to inhibit reconstitution of the computation by an untrusted party.

In addition, claim 1 further recites in part that:

- the distractive computational unit comprises a dummy computational unit;
- the computation includes a plurality of arguments;
- partitioning the computation into the plurality of computational units comprises partitioning using the Chinese Remainder Theorem (CRT);
- partitioning the computation into the plurality of computational units comprises selecting a plurality of relatively prime moduli and generating each computational unit by performing a modulo operation on each of the plurality of arguments using one of the plurality of relatively prime moduli;

- selecting the plurality of relatively prime moduli includes selecting each modulus from a superset of relatively prime moduli; and

- partitioning a plurality of computations into multiple computational units using different sets of moduli selected from the superset of relatively prime moduli.

Applicant respectfully submits that claim 1 is non-obvious over Jakobsson and Elbe and the other prior art of record. First, with respect to the concept of a distractive unit being a dummy unit, the Examiner has admitted this concept is not disclosed by Jakobsson. The Examiner instead relies on Elbe, which merely discloses the use of dummy units to be executed on different cryptographic coprocessors in a cryptographic processor, and for the purpose of inhibiting the ability to reconstruct power profiles for other coprocessors in the same processor. Applicant submits that this disclosure is insufficient to suggest modifying Jakobsson to incorporate dummy units for the purpose of inhibiting reconstruction of a computation resulting from execution of such units by an untrusted computer.

Specifically, Elbe discloses a smartcard with an integrated processor including a plurality of cryptographic coprocessors that are coupled to a bus and used to perform cryptographic computations. Elbe addresses a concern that arises as a result of the potential ability for a hacker to attack a secured processor by analyzing the "power profile" of the processor as it is performing computations (*see, e.g.,* Elbe, paragraph [0011][1]). The Examiner specifically cites paragraphs [0023], [0047] and [0049] of Elbe for allegedly disclosing dummy computations. What these passages describe, however, is an embodiment where dummy computations are performed by unused cryptographic coprocessors on the same processor chip at the same time as another cryptographic coprocessor performs a useful computation. By doing so, the power profile of the cryptographic coprocessor performing the useful computation is harder to detect, and thus, it is more difficult to reverse engineer parameters from the useful computation via its power profile.

[1] While the actual reference cited by the Examiner is WO 02/48857, which is a German language document, the Examiner apparently is citing passages from the English language equivalent to this reference, U.S. Patent Publication No. 2004/039928. Thus, for consistency, Applicant will likewise cite passages from this equivalent English language reference.

It is important to note, however, that the processor chip upon which the cryptographic coprocessors is disposed, to the extent it may be analogized as a "computer" within the context of claim 1, is effectively "trusted" from the standpoint of the smartcard system within which it exists. The processor itself, which includes all of the cryptographic coprocessors, knows which computational units are dummy units, and which are not, and is fully capable of using the results to perform desired computations. Elbe is concerned with protecting the processor from outside attacks, whereas claim 1 is concerned with protecting computations performed on an "untrusted" computers. By virtue of the computer being "untrusted" as recited in claim 1, a significant purpose of the method of claim 1 is to enable the computation to be performed without the ultimate result of the computation being determined in the untrusted computer.

From the standpoint of obviousness analysis, one of ordinary skill in the art would not look to Elbe for a suggestion that dummy computational units could be sent along with useful computational units to an untrusted computer for computation to inhibit reconstitution of computations by an untrusted party having access to the untrusted computer. Given that Elbe addresses using dummy computational units to conceal power profiles from an attack by an untrusted party on a trusted computer, Applicant submits that there is no objective reason why one of ordinary skill in the art would be motivated by Elbe to execute dummy computational units on an untrusted computer in order to prevent reconstitution of computations being performed on that computer. In fact, computations are reconstituted on the Elbe processor, so to the extent that the Elbe processor is analogous to a "computer," Elbe does nothing to inhibit reconstitution of computations on that computer.

Elbe furthermore is directed to solving a different problem, and doing so in a different way. Applicant therefore submits that Elbe does not address the shortcomings of Jakobsson, and therefore, the combination of Jakobsson and Elbe fails to disclose or suggest each and every feature of claim 1. Claim 1 is therefore non-obvious over these references.

Second, claim 1 recites partitioning a computation using the Chinese Remainder Theorem (CRT), and more specifically, partitioning a computation by performing a modulo operation on a plurality of arguments using one of a plurality of selected, relatively prime moduli. The

Examiner cites cols. 1, 4 and 5 of Jakobsson for allegedly disclosing these concepts. However, the cited passages do not address the use of either the CRT, or more generally, of modulo operations, to partition a computation.

Instead, these passages disclose computations such as DSA signature generation computations that happen to incorporate modulo operations. Put another way, the computations themselves incorporate modulo operations, but modulo operations are not used to partition a computation into multiple computational units. Claim 1, however, uses modulo operations to partition a computation into a plurality of computational units, or put another way, to determine how a computation will be partitioned into its component computational units.

In Jakobsson, computations such as DSA signature generation computations are broken up according to exponents k_i. The transformation techniques described in the reference, replication, dependency, blinding and permutation (cols. 5-8) all operate on exponents within an exponent vector G, and Applicant submits that, based upon the Examiner's interpretation of other elements of the claims, it is evident that the Examiner is interpreting the exponent vector G as the "computation" and the exponents k_i in the vector as the "computational units" for the purpose of applying Jakobsson to the claims at issue (see, e.g., col. 7, lines 60-63, "This input is denoted herein G_1 . . . and represents [sic] computational task for the DSA digital signature protocol.") However, there is no partitioning of a computational task in Jakobsson that relies on modulo operations to determine how the task is broken up into computational units. In no type of transform operation (replication, dependency, blinding and permutation) is any modulo operation ever used to select or modify an exponent in the exponent vector. In fact, for the blinding operation, the only modifications being performed on computational units rely on applying random and secret offsets (col. 6, lines 42-45). No modulo operations are ever used to either modify an exponent or determine what exponents are provided within an exponent vector.

Claim 1 recites that each computational unit is generated in connection with partitioning the computation by performing a modulo operation on each of the plurality of arguments in the computation using one of a plurality of relatively prime moduli. Claim 1 also recites that the partitioning of a computation includes selecting a plurality of relatively prime moduli.

Jakobsson, however, does not disclose performing any type of modulo operation on arguments in a computation for the purpose of partitioning the computation, nor does the reference disclose selecting relatively prime moduli. While modulo operations are disclosed in Jakobsson, it is evident from the disclosure that these modulo operations are not used to partition an operation into multiple computational units. Jakobsson therefore does not disclose or suggest this aspect of claim 1.

Furthermore, Elbe adds nothing to the rejection in this regard, as Elbe similarly fails to disclose or suggest the use of modulo operations to determine how to partition a computation into multiple computational units.

Third, claim 1 also recites the concept of partitioning a plurality of computations into multiple computational units using different sets of moduli selected from a superset of relatively prime moduli. For this concept, the Examiner relies on col. 5, lines 1-17 of Jakobsson. However, this passage discloses only that modulo operations can be performed as part of a DSA signature generation computation. There is nothing in this passage regarding selecting moduli from a superset of relatively prime moduli, doing so differently for different computations, or for selecting those moduli for the purpose of partitioning computations into computational units. As noted above, Jakobsson does not use modulo operations to partition computations, and accordingly, Applicant submits that Jakobsson cannot be interpreted to disclose or suggest selecting sets of moduli from a superset of relatively prime moduli to partition computations in the manner recited in claim 1.

Furthermore, as Elbe does not address the use of modulo operations to partition computations, Applicant submits that the combination of Elbe likewise does not suggest selecting sets of moduli from a superset of relatively prime moduli to partition computations in the manner recited in claim 1.

Accordingly, the combination of Jakobsson and Elbe does not disclose or suggest each and every feature of claim 1, and claim 1 is therefore non-obvious over the proposed combination. Applicant also submits that the Examiner has provided no objective reason why one of ordinary skill in the art would be motivated by Elbe or any other art to modify Jakobsson

to use modulo operations to partition a computation, or to do so by selecting sets of moduli from a superset of relatively prime moduli. Accordingly, claim 1 is non-obvious over the Jakobsson and Elbe. Reversal of the Examiner's rejection of claim 1, and allowance of this claim, and of claims 2-4, 6-8 and 13-16 that depend therefrom, are therefore respectfully requested.

Independent Claims 17 and 33

Next with regard to the rejection of independent claims 17 and 33, these claims recite, similar to claim 1, the concepts of using dummy computational units as distractive computational units, using the Chinese Remainder Theorem (CRT) to partition a computation, performing a modulo operation on a plurality of arguments using one of a plurality of selected, relatively prime moduli to determine how to partition a computation into multiple computational units, and using select sets of moduli from a superset of relatively prime moduli to partition computations. As discussed above in connection with claim 1, Jakobsson and Elbe do not disclose or suggest, among other features, the use of modulo operations to determine how to partition a computation into multiple computational units, or to select sets of moduli from a superset of relatively prime moduli to partition computations. Applicant therefore respectfully submits that claims 17 and 33 are non-obvious over Jakobsson and Elbe for the same reasons as presented above for claim 1. Reversal of the Examiner's rejections of independent claims 17 and 33, and allowance of these claims, as well as of claims 18-20, 22-24 and 29-32 that depend therefrom, are therefore respectfully requested.

Dependent Claims 2-3 and 18-19

Claims 2-3 and 18-19 are not argued separately.

Dependent Claims 4 and 20

Claim 4 depends from claim 2, and additionally recites that partitioning the computation uses a different algorithm than that used to partition a second computation. Claim 20 depends from claim 18 and recites similar subject matter. In rejecting these claims, the Examiner cites col. 4, lines 56-67 of Jakobsson. However, this passage merely discloses that the disclosed techniques may be used for different types of cryptographic computations. There is no disclosure or suggestion in the passage that computational units from multiple computations,

executed by the same untrusted computer, can be partitioned using different algorithms. Therefore, Applicant submits that the Examiner has failed to establish a *prima facie* case of obviousness as to claims 4 and 20, and the rejections thereof should be reversed.

Dependent Claims 6 and 22

Claims 6 and 22 are not argued separately.

Dependent Claims 7 and 23

Claim 7 depends from claim 1, and additionally recites initiating execution of at least one of the plurality of computational units on a second computer. Claim 23 depends from claim 17 and recites similar subject matter. In rejecting these claims, the Examiner cites col. 3, lines 30-45. However, this passage merely discloses that the disclosed techniques may be used on different types of computers. There is no disclosure or suggestion in the passage that different computational units from the same computation may be executed on different computers. Therefore, Applicant submits that the Examiner has failed to establish a *prima facie* case of obviousness as to claims 7 and 23, and the rejections thereof should be reversed.

Dependent Claims 8 and 24

Claims 8 and 24 are not argued separately.

Dependent Claims 13-14 and 29-30

Claim 13 depends from claim 1, and additionally recites that the untrusted computer is coupled to a grid computing network. Claim 29 depends from claim 17 and recites similar subject matter. Claims 14 and 30 respectively depend from claims 13 and 29. In rejecting these claims, the Examiner cites passages at col. 3 of Jakobsson.

Nowhere in Jakobsson, however, is the concept of a computer grid ever discussed. A grid computing network is not merely a collection of servers, as is disclosed in Jakobsson. Applicant clearly defines a grid computing network in the Application, e.g., at page 1, line 13 to page 2, line 2, as follows:

> One such computational utility environment is referred to as grid computing, where a collection of potentially heterogeneous and geographically remote computer systems are connected together in a distributed fashion to share their respective resources and ***present the appearance of a single large and powerful virtual computer system***. A grid computing environment may be used to share various hardware and software resources such as processors, applications, storage, memory, printers, network connections, and other peripheral devices. ***In a computational grid, the hardware and/or software resources of multiple computers are abstracted, with specialized software used to pass work to various resources in such a manner as to maximize the utilization of the underlying resources***. (*emphasis added*).

Jakobsson therefore fails to disclose or suggest a grid computing network, as that term should be interpreted for the purposes of Applicant's claims. Likewise, Elbe is used in a smartcard application, and thus fails to suggest a grid computing network as well. Therefore, Applicant submits that the Examiner has failed to establish a *prima facie* case of obviousness as to claims 13-14 and 29-30, and the rejections thereof should be reversed.

Dependent Claims 15 and 31

Claim 15 depends from claim 13, and additionally recites that partitioning the computation is performed by a broker computer coupled to the grid computing network. The method also recites receiving the computation from a client computer. Claim 31 depends from claim 29 and recites similar subject matter. In rejecting these claims, the Examiner cites col. 4, lines 20-31 and Figs. 1 and 3 of Jakobsson. However, these passages disclose an originator and a pool of servers, but no additional devices that participate in computations. The originator disclosed in Jakobsson could perhaps be analogized to a broker or a client; however, the Examiner apparently would use this same element for both a broker and a client. Doing so, however, would effectively read out the inclusion of both elements from the claims, and thus would be improper. Accordingly, in addition to being distinguishable from Jakobsson and Elbe for being directed to a grid computing network, these claims are additionally distinguishable for incorporating a client and a broker in addition to a grid computing network. Therefore, Applicant submits that the Examiner has failed to establish a *prima facie* case of obviousness as to claims 15 and 31, and the rejections thereof should be reversed.

Dependent Claims 16 and 32

Claims 16 and 32 are not argued separately.

CONCLUSION

Applicant respectfully requests that the Board reverse the Examiner's rejections of claims 1-4, 6-8, 13-20, 22-24 and 29-33, and that the Application be passed to issue. If there are any questions regarding the foregoing, please contact the undersigned at 513/241-2324. If any other charges or credits are necessary to complete this communication, please apply them to Deposit Account 23-3000.

Respectfully submitted,

March 28, 2008
Date

/Scott A. Stinebruner/
Scott A. Stinebruner
Reg. No. 38,323
WOOD, HERRON & EVANS, L.L.P.
2700 Carew Tower
441 Vine Street
Cincinnati, Ohio 45202
Telephone: (513) 241-2324
Facsimile: (513) 241-6234

VIII. CLAIMS APPENDIX: CLAIMS ON APPEAL (S/N 10/760,592)

Listing of Claims:

1. (Previously Presented) A method of initiating performance of a computation on at least one untrusted computer, the method comprising:

partitioning the computation into a plurality of computational units that are combinable to generate a result for the computation, wherein the computation includes a plurality of arguments, wherein partitioning the computation into the plurality of computational units comprises partitioning using the Chinese Remainder Theorem (CRT), wherein partitioning the computation into the plurality of computational units comprises selecting a plurality of relatively prime moduli and generating each computational unit by performing a modulo operation on each of the plurality of arguments using one of the plurality of relatively prime moduli, and wherein selecting the plurality of relatively prime moduli includes selecting each modulus from a superset of relatively prime moduli;

partitioning a plurality of computations into multiple computational units using different sets of moduli selected from the superset of relatively prime moduli;

generating at least one distractive computational unit, wherein the distractive computational unit comprises a dummy computational unit;

initiating execution of both the at least one distractive computational unit and computational units from multiple computations on the untrusted computer to inhibit reconstitution of the computations by an untrusted party;

receiving result data generated during execution of the computational units from the multiple computations; and

generating results for the multiple computations from the result data.

2. (Original) The method of claim 1, wherein the distractive computational unit comprises a computational unit generated from partitioning a second computation.

3. (Original) The method of claim 2, wherein initiating execution of both the at least one distractive computational unit and at least one of the plurality of computational units includes interleaving the at least one distractive computational unit among multiple computational units from the plurality of computational units.

4. (Original) The method of claim 2, wherein partitioning the computation uses a different algorithm than that used to partition the second computation.

5. (Canceled).

6. (Original) The method of claim 1, wherein the distractive computational unit comprises a computational unit generated from a second partitioning of the computation.

7. (Original) The method of claim 1, further comprising initiating execution of at least one of the plurality of computational units on a second computer.

8. (Original) The method of claim 1, further comprising initiating execution of all of the plurality of computational units on the untrusted computer.

9.-12. (Canceled).

13. (Original) The method of claim 1, wherein the untrusted computer is coupled to a grid computing network.

14. (Original) The method of claim 13, wherein partitioning the computation is performed by a client computer coupled to the grid computing network.

15. (Original) The method of claim13, wherein partitioning the computation is performed by a broker computer coupled to the grid computing network, the method further comprising receiving the computation from a client computer.

16. (Original) The method of claim 1, wherein partitioning the computation, generating the distractive computational unit, and initiating execution of both the distractive computational unit and the one of the plurality of computational units on the untrusted computer are performed by at least one computer coupled to the untrusted computer, the method further comprising communicating the distractive computational unit and the one of the plurality of computational units to the untrusted computer.

17. (Previously Presented) An apparatus, comprising:

at least one processor; and

program code configured to be executed by the at least one processor to initiate performance of a computation on at least one untrusted computer by partitioning the computation into a plurality of computational units that are combinable to generate a result for the computation, generating at least one distractive computational unit, and initiating execution of both the at least one distractive computational unit and at least one of the plurality of computational units on the untrusted computer to inhibit reconstitution of the computation by an untrusted party;

wherein the program code is configured to partition the computation into the plurality of computational units using the Chinese Remainder Theorem (CRT), wherein the computation includes a plurality of arguments, wherein the program code is configured to partition the computation into the plurality of computational units by selecting a plurality of relatively prime moduli, and generating each computational unit by performing a modulo operation on each of the plurality of arguments using one of the plurality of relatively prime moduli, wherein the program code is configured to select the plurality of relatively prime moduli from a superset of relatively prime moduli, wherein the program code is further configured to partition a plurality of computations into multiple computational units using different sets of moduli selected from the

superset of relatively prime moduli, and initiate execution of computational units from multiple computations on the untrusted computer, wherein the distractive computational unit comprises a dummy computational unit, and wherein the program code is further configured to receive result data generated during execution of the computational units from the multiple computations and generate results for the multiple computations from the result data.

18. (Original) The apparatus of claim 17, wherein the distractive computational unit comprises a computational unit generated from partitioning a second computation.

19. (Original) The apparatus of claim 18, wherein the program code is configured to initiate execution of both the at least one distractive computational unit and at least one of the plurality of computational units by interleaving the at least one distractive computational unit among multiple computational units from the plurality of computational units.

20. (Original) The apparatus of claim 18, wherein the program code is configured to partition the computation using a different algorithm than that used to partition the second computation.

21. (Canceled).

22. (Original) The apparatus of claim 17, wherein the distractive computational unit comprises a computational unit generated from a second partitioning of the computation.

23. (Original) The apparatus of claim 17, wherein the program code is further configured to initiate execution of at least one of the plurality of computational units on a second computer.

24. (Original) The apparatus of claim 17, wherein the program code is further configured to initiate execution of all of the plurality of computational units on the untrusted computer.

25.-28. (Canceled).

29. (Original) The apparatus of claim 17, wherein the untrusted computer is coupled to a grid computing network.

30. (Original) The apparatus of claim 29, further comprising a client computer coupled to the grid computing network and upon which the program code resides.

31. (Original) The apparatus of claim29, further comprising a client computer coupled to the grid computing network and upon which the program code resides, wherein the program code is further configured to receive the computation from a client computer.

32. (Original) The apparatus of claim 17, wherein the program code resides on a separate computer coupled to the untrusted computer, and wherein the program code is further configured to communicate the distractive computational unit and the one of the plurality of computational units to the untrusted computer.

33. (Previously Presented) A program product, comprising:

program code configured to initiate performance of a computation on at least one untrusted computer by partitioning the computation into a plurality of computational units that are combinable to generate a result for the computation, generating at least one distractive computational unit, and initiating execution of both the at least one distractive computational unit and at least one of the plurality of computational units on the untrusted computer to inhibit reconstitution of the computation by an untrusted party; and

a computer readable recordable storage medium bearing the program code;

wherein the program code is configured to partition the computation into the plurality of computational units using the Chinese Remainder Theorem (CRT), wherein the computation includes a plurality of arguments, wherein the program code is configured to partition the computation into the plurality of computational units by selecting a plurality of relatively prime moduli, and generating each computational unit by performing a modulo operation on each of the plurality of arguments using one of the plurality of relatively prime moduli, wherein the program code is configured to select the plurality of relatively prime moduli from a superset of relatively prime moduli, wherein the program code is further configured to partition a plurality of computations into multiple computational units using different sets of moduli selected from the superset of relatively prime moduli, and initiate execution of computational units from multiple computations on the untrusted computer, and wherein the distractive computational unit comprises a dummy computational unit, and wherein the program code is further configured to receive result data generated during execution of the computational units from the multiple computations and generate results for the multiple computations from the result data.

34. (Canceled).

IX. EVIDENCE APPENDIX

[10/760,592]

None.

X. RELATED PROCEEDINGS APPENDIX

[10/760,592]

None.